COREELARTION COEFFICIENTS

CORRELATION TYPES

There are several types of correlation metrics that can be used for different types of data: The most popular are

- **Pearson's coefficient** that measures linear correlation in numerical data
- **and Spearman or Kendall coefficients used to** compare the ranks of data

Correlation Technique	Relationship	Datatype of features
Pearson	Linear	Quantitative and Quantitative
Spearman	Non-linear	Ordinal and Ordinal
Point-biserial	Linear	Binary and Quantitative
Cramer's V	Non-linear	Categorical and Categorical
Kendall's tau	Non-linear	Two Categorical or Two Quantitative

SciPy CODE EXAMPLES

All Scipy Functions Return
-> corr. coefficient
-> *p-value*

```
# create example data
>>> x = np.arange(10, 20)
>>> y = np.array([2, 1, 4, 5, 8, 12, 18, 25, 96, 48])

# Pearson's r
>>> scipy.stats.pearsonr(x, y)
(0.7586402890911869, 0.010964341301680832)

# Spearman's rho
>>> scipy.stats.spearmanr(x, y)
SpearmanrResult( correlation=0.975757, pvalue=1.4675461877e-06)

# Kendall's tau
>>> scipy.stats.kendalltau(x, y)
KendalltauResult(correlation=0.911111, pvalue=2.976190462e-05)
```

CAUTION

Check NA policy in each function

INTERPRETATION

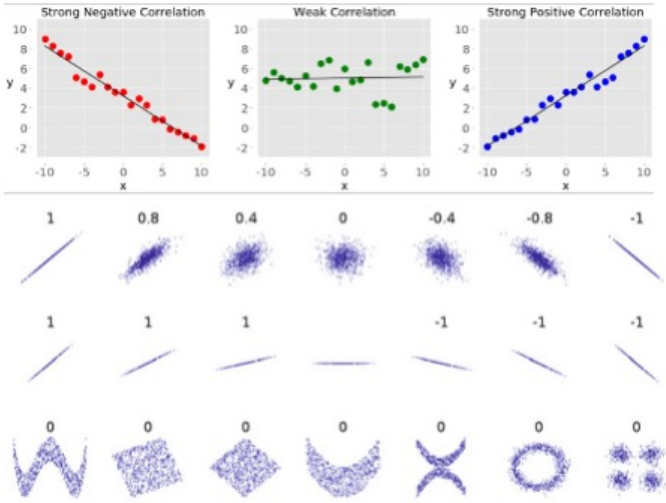


Fig. 11.1 Several sets of (x, y) points, with the correlation coefficient of x and y for each set. Note that the correlation reflects the noisiness and direction of a linear relationship (*top row*), but not the slope of that relationship (*middle*), nor many aspects of nonlinear relationships (*bottom*). N.B.: the figure in the center has a slope of 0 but in that case the correlation coefficient is undefined because the variance of Y is zero. (In Wikipedia. Retrieved May 27, 2015, from http://en.wikipedia.org/wiki/Correlation_and_dependence.)

PEARSON R *CORRELATION COEEFICIENT*

The *correlation coefficient* between two variables answers the question: "Are the two variables related? That is, if one variable changes, does the other also change?" If the two variables are normally distributed, the standard measure of determining the correlation coefficient, often ascribed to Pearson, is

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \quad (11.1)$$

With the sample covariance s_{xy} defined as

$$s_{xy} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{n-1} \quad (11.2)$$

and s_x, s_y the sample standard deviations of the x and y values, respectively, Eq. 11.1 can also be written as

$$r = \frac{s_{xy}}{s_x \cdot s_y}. \quad (11.3)$$

Pearson's correlation coefficient, sometimes also referred to as *population correlation coefficient* or *sample correlation*, can take any value from −1 to +1. Examples are given in Fig. 11.1. Note that the formula for the correlation coefficient is symmetrical between x and y—which is not the case for linear regression!

REPORTING RESULTS

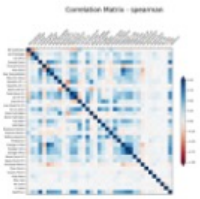
Reporting individual Results:

- … and … were found to be moderately positively corr., $r(38) = .34$, $p = .032$.
- …. were found to be strongly correlated, $r(128) = .89$, $p < .01$.
- …. were negatively correlated, $r(78) = -.45$, $p < .001$
- no linear correlation was found, between ..&…, $r(38) = .02$, $p = .005$

Comments

- Degrees of freedom for *r* is *N* - 2 – denoted in brackets -> r(120)
- Report the exact *p*value, + state your alpha, eg 0.05 level early in your results
- *r* statistic should be stated at 2, or 3 decimal places

CORRELATION MATRIX & TRENDLINE



Pandas

```
>>> df_corr = df.corr(method='pearson')
>>> fig, ax = plt.subplots()
>>> ax.matshow(
        df_corr, cmap=cmap, vmin= -1, vmax=1)
    # min,max values will scale the heatmap
```

Scipy

```
>>> rho, pvalues = scipy.stats.spearmanr(np.array)
```

Obtained with linear regr. In scipy

```
>>> result = scipy.stats.linregress(x, y)
    result.slope        # 7.4363636363636365
    result.intercept    # -85.92727272727274
    result.rvalue       # 0.7586402890911869
    result.pvalue       # 0.010964341301680825
    result.stderr       # 2.257878767543913
```

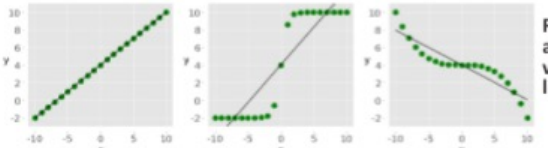
RANK CORRELATION

- ***For Not normally distributed data***
- ***For Non-linear relationships between compared var's***

SPEARMAN RHO

The **Spearman correlation coefficient** Is calculated the same way as the Pearson correlation coefficient but takes into account their ranks instead of their values.

- Denoted with the Greek letter rho (ρ) and called **Spearman's rho**.
- $-1 \le \rho \le 1$.
  - If ρ=1 => the function between x and y increases monotonically
  - If ρ=1 => the function between x and y decreases monotonically



Rank correlation, allows comparing variables with non-linear relationships

Sopearman Rho uses the ranks or the orderings of data points in compared variables/features, If the orderings are similar, then the correlation is strong, positive, and high. However, if the orderings are close to reversed, then the correlation is strong, negative, and low. In other words, rank correlation is concerned only with the order of values, not with the particular values from the dataset.

KENDAL TAU

Kendal Tau CC is harder to calculate than Spearman's but its confidence intervals are more reliable

- **Kendall correlation coefficient** is calculated as the difference in the counts of concordant and discordant ranked data pairs, relative to the total number of x-y pairs in compared rankings.
- VALUES:
  - $-1 \le \tau \le 1$.
  - $\tau = 1$ - the ranks of the corresponding values in **x** and **y** are the same. In other words, all pairs are concordant.
  - $\tau = -1$ - the rankings in **x** are the reverse of the rankings in **y**. In other words, all pairs are discordant.

HOW IT IS DONE

- Lets, have two random variables x, & y, that we wish to compare
- All values in x, and y were ranked independently.
- Now, if we compare all pairs of values in x, and y, using the same pairs in xi:xj, and yi:xj in both variables at each comparison, we will label the results in one the 3 following classes:
  - **Pair of points has concordant ranks in x,y;** $(x_i > x_j \text{ and } y_i > y_j)$ *or* $(x_i < x_j \text{ and } y_i < y_j)$
  - **… has discordant** ranks in x,& y ; $(x_i < x_j \text{ and } y_i > y_j)$ *or* $(x_i > x_j \text{ and } y_i < y_j)$
  - **… the same ranks in x&y (tie);** *a tie in x* $(x_i = x_j)$ *or a tie in y* $(y_i = y_j)$*, or in both,*

According to the `scipy.stats` official docs, the Kendall correlation coefficient is calculated as $\tau = (n^+ - n^-) / \sqrt{((n^+ + n^- + n^x)(n^+ + n^- + n^y))}$, where:

- n^+ is the number of concordant pairs
- n^- is the number of discordant pairs
- n^x is the number of ties only in **x**
- n^y is the number of ties only in **y**

Greek letter tau (τ)

If a tie occurs in both **x** and **y**, then it's not included in either n^x or n^y.

KENDAL TAU – DIFFERENT TYPES

- **scipy.stats.kendalltau has a.&b, varinats, that treat ties differently, b is default**
- **additionally, scipy offers weighted kendal tau,** in which exchanges of high weight are more influential than exchanges of low weight.